IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

REMBRANDT PATENT INNOVATIONS, LLC and REMBRANDT SECURE COMPUTING, LP,	)	Civil Action No. 2:14-cv-15
Plaintiffs,	)	**JURY TRIAL DEMANDED**
v.	)	
APPLE INC.,	)	
Defendant.	)	

REMBRANDT PATENT INNOVATIONS, LLC'S AND REMBRANDT SECURE COMPUTING, LP'S OPENING CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

I. INTRODUCTION

Rembrandt's proposed constructions for disputed terms in U.S. Patent No. 6,185,678 (Exhibit A) follow the Federal Circuit's framework for claim construction, stay true to the claim language and intrinsic evidence, and reflect the language the inventors used to describe their invention. Rembrandt respectfully requests the Court adopt its proposed constructions.

II. THE ’678 PATENT

The ’678 Patent was filed on October 2, 1998, claiming the benefit of a provisional application filed October 2, 1997. The inventors of the ’678 Patent are Drs. Arbaugh, Farber, Keromytis, and Smith, and the invention reflects their work at the University of Pennsylvania, some of which also contributed to Dr. Arbaugh's Ph.D. thesis.

One of the first actions a computer performs when it is turned on is to load a software component, called a “boot loader” or a “bootstrap loader,” which is used to load other software the computer needs to operate. The term is reminiscent of the expression “pulling oneself up by the bootstraps.” The process of loading this software is called “booting up,” “boot,” or “booting,” the system.

During the boot process, the computer system is vulnerable to unauthorized or malicious software. The ’678 Patent describes and claims techniques for booting a computer system securely and recovering software components that do not pass security measures. *See* Ex. A at 4:35-43.

Figure 1a in the ’678 Patent shows the boot process for a typical IBM PC architecture, which includes several software components called “boot components” or “bootstrap components”: system BIOS 112, expansion ROMs 122, boot block 132, and operating system kernel 142.[1] *See, e.g.*, Ex. A at 1:53-57. The boot components are software organized in functional layers demarcated, for example, by dotted lines in Figure 1a (below). Computer systems "are organized as layers to limit complexity" and "[a] common layering principle is the use of layers of abstraction to mark layer boundaries." Ex. A at 1:27-29. In Figure 1a, the boot process and boot components are implemented using four software layers, **110**, **120**, **130**, and **140**. Ex. A at 7:56-60.

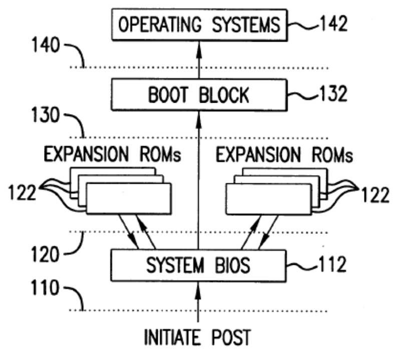

FIG. 1a

First layer **110** includes system BIOS **112** and corresponds to the first phase of the bootstrap process. Ex. A at 7:61-62. The system BIOS may pass control to software in one or more optional expansion ROMs **122** in second layer **120**. Ex. A at 8:12-22. Once the software on the expansion ROMs finishes executing, system BIOS passes control to third layer **130**, which contains boot block **132**. Ex. A at 8:23-29. Boot block **132** eventually passes control to fourth layer **140**, which contains the operating system kernel **142**. Ex. A at 8:29-31. The operating system kernel loads the remainder of the operating system.

[1] The operating system is software that provides various system services and utilities including, for example, a file system, networking support, and memory management, among other things. The operating system kernel is a central component of the operating system and is used to transfer control to the operating system itself.

One attempt to provide a secure computing environment added dedicated security hardware, such as a cryptographic coprocessor or a PCMCIA card, to the system. Ex. A at 2:1-32. Other attempts simply assumed that some or all of the boot components were trusted, and focused their energy elsewhere. Ex. A at 2:33-47. But as the patent notes, the operating system cannot be trusted if it is invoked by an untrusted process. *See* Ex. A at 1:39-44. If one loads the operating system using malicious or infected software layers, something other than the intended operating system may load as a result. Prior attempts to provide a secure computing environment also failed to provide a satisfactory recovery option if the boot process failed. The inventors sought to overcome deficiencies in prior-art security systems and methods.

Systems and methods using the present invention construct a "chain of integrity checks," beginning at power-on and continuing through the transfer of control from the bootstrap components to the operating system kernel. Ex. A at 4:40-43. For example, the '678 Patent explains that transitions from one layer to the next occur only after verifying the next layer's integrity. *See* Ex. A at 1:34-39, 4:35-43, 8:60-67, 9:35-10:3. The patent provides methods of recovery when a boot component fails its integrity check.

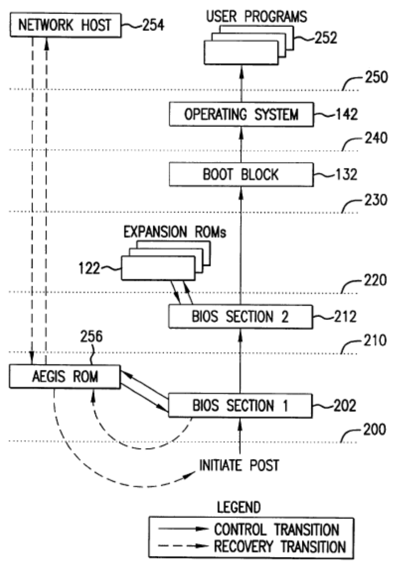

FIG. 2a

As shown in Figure 2a, the ’678 Patent teaches splitting the system BIOS into separate layers **200** and **210** containing BIOS Section 1 (**202**) and BIOS Section 2 (**212**). Layer **200** contains a small section of trusted software, BIOS Section 1, whose integrity is assumed to be valid, Ex. A at 8:45-49, but still may be tested to detect inadvertent errors. Ex. A at 8:49-51. The second layer **210** of the BIOS contains the rest of the BIOS functionality, and is not trusted until it is verified. Ex. A at 8:51-52, 9:35-39.

Following the first layer **200**, a transition from one layer to the next does not occur unless the integrity of the next layer is first verified. Ex. A at 4:35-43, 8:60-67. Layers are verified using public key cryptography and a cryptographic hash function. *Id.* Verification involves computing a cryptographic hash value of the software being verified, and comparing the computed cryptographic hash value with one obtained from a digital signature associated with the software. *See, e.g.*, Ex. A at 9:35-39. Control transfers to the next software layer only if the computed cryptographic hash value matches the cryptographic hash value obtained from the signature. *See, e.g.*, *id.*

The ’678 Patent describes a recovery mechanism that allows the computer system to recover a software component that failed its integrity check. Ex. A at 10:1-6. For example, the patent teaches that a copy of the component may be recovered through a trusted repository. Ex. A at 10:56-67. The “trusted repository” can either be an expansion ROM that contains copies of the software component, or it can be one or more network hosts 254 that allow the system to obtain copies of the software component that failed its integrity check. *See, e.g.*, Ex. A at 10:44-67. If the trusted repository is a network host, the recovery process can use a secure protocol to obtain a copy of the software component that failed its integrity check. Ex. A at 4:48-51. The secure protocol combines cryptographic algorithms with a protocol to add security to the recovery process. *See, e.g.*, Ex. A 4:56-57. For example, in one embodiment, the secure protocol can be based on adding authentication to the Dynamic Host Configuration Protocol (DHCP) and Trivial File Transfer Protocol (TFTP). Ex. A at 19:36-38. Authentication is added to DHCP and TFTP by adding SHA1 message digests (cryptographic hash values) to DHCP and TFTP messages. *See* Ex. A at 15:50-67, 18:24-27, 19:38-45.

III. LEGAL PRINCIPLES

Claim construction is a question of law. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 977-79 (Fed. Cir. 1995) (en banc). Claim terms “are generally given their ordinary and customary meaning,” which “is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005) (en banc) (internal citation omitted). There are only two exceptions to this general rule: 1) when a patentee sets out a definition and acts as his own lexicographer, or 2) when the patentee disavows the full scope of a claim term either in the specification or during prosecution. *Thorner v. Sony Computer Entm’t Am., LLC,* 669 F.3d 1362, 1365 (Fed. Cir. 2012). Both exceptions require a clear and explicit statement by the patentee. *Id.* at 1367-68. To constitute disclaimer, for example, there must be a clear and unmistakable disclaimer. *Id.* at 1366-67. And to act as its own lexicographer, a patentee must "clearly set forth a definition of the disputed claim term" other than its plain and ordinary meaning. *Id.* at 1365.

A person of ordinary skill in the art "is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Phillips*, 415 F.3d at 1313. Proper claim construction relies primarily on an analysis of the patent's intrinsic evidence, i.e., the language of the claims, the specification, and the prosecution history. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582-83 (Fed. Cir. 1996).

The proper construction for a claim term must align with what the patent describes as the invention and what the inventors actually invented. *Phillips*, 415 F.3d at 1316. The patent's specification "is always highly relevant" to the analysis, as it "necessarily informs the proper construction of the claims." *Id.* at 1315-16. "Usually, [the specification] is dispositive; it is the single best guide to the meaning of a disputed term." *Vitronics*, 90 F.3d at 1582. The specification also helps resolve ambiguous claim terms if "the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002).

It is improper to "read unstated limitations into claim language." *Rambus Inc. v. Infineon Techs. Ag*, 318 F.3d 1081, 1088 (Fed. Cir. 2003); *see also Thorner*, 669 F.3d at 1366-67. "We do not import limitations into claims from examples or embodiments appearing only in a patent's written description, even when a specification describes very specific embodiments of the invention or even describes only a single embodiment." *JVW Enters., Inc. v. Interact Accessories, Inc.*, 424 F.3d 1324, 1335 (Fed. Cir. 2005).

A court may also consider extrinsic evidence, but such evidence is less reliable and less significant than the intrinsic record. *Phillips*, 415 F.3d at 1317-18. A court should discount extrinsic evidence at odds with the intrinsic record. *Id.* at 1318.

IV. CLAIM CONSTRUCTIONS OF THE DISPUTED TERMS

A. "coupled to"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"coupled to" [Claims 1, 3]	"connected directly or indirectly"	"directly linked to"

The parties' dispute concerns whether "coupled to" encompasses indirect connections. The language of the claims and the doctrine of claim differentiation support Rembrandt's proposed construction. For example, claim 1 recites "a trusted repository *coupled to* said expansion bus."[2] Ex. A at 22:7 (emphasis added). Claim 3 depends from claim 1 and provides that the trusted repository may be "a host computer communicating with said computer system through a communications interface *coupled to* said expansion bus." Ex. A at 22:15-18 (emphasis added). Claim 3 recites the "coupling" of the trusted repository to the expansion bus through a "communications interface *coupled to* said expansion bus" (emphasis added), so the communications interface provides an indirect connection between the trusted repository and the expansion bus. The term "coupled to" in independent claim 1 must include indirect connections to be broader than its dependent claim 3.

[2] Claim 1 recites "coupled to" four times, but never requires limiting this term to a direct connection. Claim 1's recitation of "a trusted repository *coupled to* said expansion bus" illustrates this point.

Figure 1c, excerpted below, also supports Rembrandt's proposed construction:

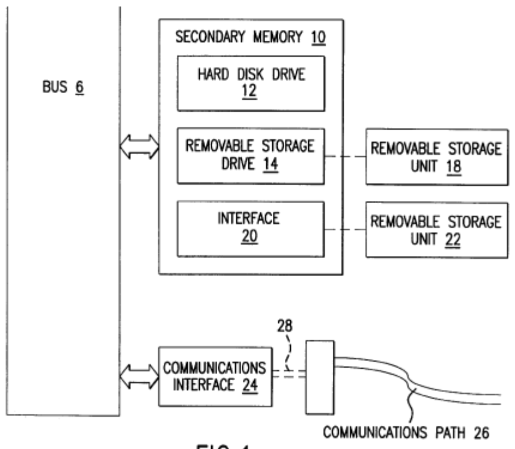
BUS 6
SECONDARY MEMORY 10
HARD DISK DRIVE 12
REMOVABLE STORAGE DRIVE 14
REMOVABLE STORAGE UNIT 18
INTERFACE 20
REMOVABLE STORAGE UNIT 22
28
COMMUNICATIONS INTERFACE 24
COMMUNICATIONS PATH 26

FIG.1c

Figure 1c depicts "communications interface 24" coupled to the expansion bus. "Communications interface 24 allows software and data to be transferred between computer system 1 and external devices," and may be "a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, etc." Ex. A at 7:7-12. In this case, the trusted repository would not be directly connected to the expansion bus, but would instead be indirectly connected to that bus via communications path 26 and communications interface 24, consistent with Rembrandt's construction.

The Federal Circuit has also construed "coupled" as Rembrandt proposes, noting it is a "broad and general term." *Johnson Worldwide Assocs., Inc. v. Zebco Corp.*, 175 F.3d 985, 992 (Fed. Cir. 1999). The Federal Circuit addressed a situation similar to this one in *Bradford Co. v. Conteyor North America, Inc.*, 603 F.3d 1262 (Fed. Cir. 2010). There, claim 1 recited that a "dunnage structure" of a claimed container was "coupled to" the frame of the container. *Id.* at 1270-71. Dependent claim 10 required the container to have rails coupled both to the container and to the dunnage structure. *Id.* The Federal Circuit rejected the defendant's narrow construction of "coupled" as requiring a direct connection, and instead held: "the doctrine of claim differentiation supports a reading that allows an indirect coupling of the dunnage to the frame of the container." *Id.* at 1271. "In light of a dependent claim that clearly states an indirect attachment of the dunnage structure, the scope of independent claim 1 is presumed to be broader to allow for other types of indirect attachments." *Id.* As in *Bradford*, dependent claim 3 in the '678 Patent recites an indirect coupling of a "trusted repository" to the "expansion bus," so "coupled to" in the independent claim should also permit indirect connection. *Id.* Nothing in the specification indicates that the patentees intended otherwise. *See also Negotiated Data Solutions, LLC v. Dell, Inc.*, 596 F. Supp. 2d 949, 963-64 (E.D. Tex. 2009) (holding that "coupled" means more than a "direct physical connection," and defining "coupled" to mean "connected directly or indirectly.").

Neither the claims nor the intrinsic evidence supports Apple's proposed construction, limiting "coupled to" to a direct connection. For example, Apple's proposed construction is inconsistent with at least claim 3 and Figure 1c. Such a construction is not correct. *Wright Med. Tech., Inc. v. Osteonics Corp.*, 122 F.3d 1440, 1445 (Fed. Cir. 1997) ("[W]e must not interpret an independent claim in a way that is inconsistent with a claim which depends from it . . . ."). Apple's proposed construction should be rejected, and the Court should adopt Rembrandt's proposed construction.

B. "boot component"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"boot component" [Claims 1, 4, 7]	"one or more software layers used in computer initialization"	"a module used by the system BIOS to initialize the computer system"

The parties agree that a claimed "boot component" is used in computer initialization, but they disagree whether a "boot component" is software or may be a more general "module," that may be something besides software. The parties also dispute whether a “boot component” must be “used by the system BIOS.”

Apple even acknowledges that the “boot components” must be software. Apple’s proposed construction of “trusted repository” is “an expansion ROM or network host that . . . contains copies of the plurality of boot components.” To contain “copies of the plurality of boot components,” the boot components must be software.

The claim language shows that a “boot component” is software. Claim 1 recites recovering a boot component through a trusted repository if the integrity verification of a boot component fails. Ex. A at 22:8-10 (“means for verifying the integrity of said boot components and said system BIOS wherein integrity failures are recovered through said trusted repository.”). As explained below, the trusted repository may have software.

The specification also ties a “boot component” to layers of software, or “code.” Ex. A at 6:7-14 (“AEGIS increases the security of the boot process by ensuring the integrity of bootstrap code.”), 8:23-31 (referring to “bootstrap code” in the BIOS and “code contained in boot sector”), 9:40-10:1 (referring to “bootstrap code”). The specification describes the boot components as software “layers” corresponding to the system BIOS (one layer 110 in Figure 1a, two layers 200 and 210 in Figure 2a), optional expansion ROMs, a boot block (or boot sector), and operating system kernel. Ex. A at 1:54-57; Figs. 1a & 2a. Figure 2a teaches that a “boot component” is one or more software layers, as Rembrandt proposes. The prosecution history is in accord. During prosecution, the patentees explained the “boot process” begins with power-on to the processor and ends when execution control is passed to an operating system kernel. Ex. B at REMBAPPL00021704. “The software components involved in this process are (1) the system BIOS, usually located on Flash ROM on the motherboard; (2) expansion ROM’s usually located on expansion boards on the system IO bus; (3) the operating system boot block; and (4) the operating system kernel." Ex. B at REMBAPPL00021704-05. These software components correspond to the examples of "boot components" in the specification implemented in the software "layers" depicted in Figures 1a and 2a.

The term "module" in Apple's construction is no more precise than "component," so it adds nothing to the construction. *Power-One, Inc. v. Artesyn Techs., Inc.*, 599 F.3d 1343, 1348 (Fed. Cir. 2010) ("The terms, as construed by the court, must 'ensure that the jury fully understands the court's claim construction rulings and what the patentee covered by the claims.'" (citing *Sulzer Textil A.G. v. Picanol N.V.*, 358 F.3d 1356, 1366 (Fed. Cir. 2004))). To the extent "module" means something other than software, it has no support. *See, e.g.*, Ex. A at 6:33 (using the term "module" to refer to "replacement" software obtained from a trusted repository).

Apple's proposed construction requires all boot components be "used by the system BIOS." Such a limitation finds no support in the specification. This conflicts with the specification because the "operating system kernel" is a boot component, Ex. A at 1:54-57, but it is never "used by the system BIOS." *See, e.g.*, Ex. A at Fig. 2a (showing that operating system kernel 142 is "used by" boot block 132).

C. "system BIOS"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"system BIOS" [Claims 1, 4]	"one or more layers of software that perform startup operations, including initializing hardware and transferring control to a layer that loads the operating system kernel"	"firmware that controls the computer system from startup until the transfer of control to a boot component that loads the operating system kernel"

The parties' proposed constructions for "system BIOS" substantially overlap. There appears to be no dispute that the claimed "system BIOS" controls the computer system from startup, and therefore performs startup operations including initializing hardware, until it transfers control to different software (a boot component) that loads the operating system kernel. The crux of the parties' dispute is whether the claimed "system BIOS" consists of one or more "software" layers (such as Figures 1a and 2a show) or must be "firmware." The intrinsic evidence requires the system BIOS be layers of software, not firmware.

The specification describes the system BIOS as "software," and is not limited to only "firmware" as Apple proposes. In the context of Figure 2a, for example, the patent describes the system BIOS divided over two software "layers" 200 and 210. Ex. A at 8:39-52. The first layer 200 of system BIOS is described as containing "trusted software," Ex. A at 8:45-46, and the second layer 210 containing "the remainder of the usual system BIOS code," Ex. A at 8:51-52. *See also* Ex. A at 8:45-52, 9:12-21; 9:33-65, Fig. 2a.

During prosecution, the patentees characterized the "system BIOS" in the '678 Patent as a "software component" used in the boot process:

> The "boot process" ends when execution control is passed to an operating system kernel. The software components involved in this process are (1) the system BIOS, usually located on Flash ROM on the motherboard; (2) expansion ROM's usually located on expansion boards on the system IO bus; (3) the operating system boot block; and (4) the operating system kernel.

Ex. B at REMBAPPL00021704-05.

The specification and prosecution history do not contain any clear and explicit statements or disclaimers limiting "system BIOS" to firmware implementations, as Apple proposes. *See, e.g., Thorner*, 669 F.3d at 1365-66. The Court should adopt Rembrandt's proposed construction.

D. "a trusted repository"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"a trusted repository" [Claims 1, 3, 7]	"an expansion ROM or one or more network hosts that are assumed to operate as expected by the computer system"	"an expansion ROM or network host that is assumed to operate as expected by the system and contains copies of the plurality of boot components"

The parties agree that a "trusted repository" includes either an expansion ROM or a network host. *See, e.g.*, Ex. A at 10:44-46. The parties also agree that a repository is trusted when it is "assumed to operate as expected by the" computer system. The parties dispute, however, whether the trusted repository may include more than one network host, and whether the "trusted repository" must contain copies of the boot components.

Consistent with Rembrandt's proposed construction, the specification discloses a preferred embodiment including "network hosts 254." *See* Ex. A at 8:57-59; Fig. 2a. The disclosed "network hosts" is inconsistent with Apple's proposed construction.

In another embodiment, the computer system recovers a component that has failed its integrity check by first contacting a DHCP server (network host) that provides the address of a second server (another network host) from which to recover the component. Ex. A at 16:54-59. The DHCP server "provide[s] the name and server location of a bootstrap program" to the client, which "uses TFTP . . . to contact [that] server and transfer the file." Ex. A at 16:55-59. Figure 3 shows the format of a DHCP message, which allocates space for the IP address of the "next server." The specification thus contemplates that the "trusted repository" may comprise multiple servers, and supports Rembrandt's proposed construction that the "trusted repository" covers "one or more network hosts." *See* Ex. A at 16:55-59; *infra* Part IV.H (explaining that the network host is a server); *see also* Ex. A at 16:48 ("DHCP servers 420"), 17:32 ("Server 420 (Trusted Repository)").

The intrinsic record does not require the trusted repository to contain copies of the boot components. Claim 1 only requires that integrity failures are "recovered through said trusted repository" not that the trusted repository contains copies of each boot components. Apple cannot read limitations into the claims absent an express definition or clear and unmistakable disclaimer in the specification or prosecution history, which does not exist with respect to the "trusted repository" in the '678 Patent. *See, e.g.*, *Thorner*, 669 F.3d at 1365-66.

The specification describes trusted repositories that contain copies of the boot components, but all of these embodiments use an expansion ROM, not a network host, as the trusted repository. Ex. A at 10:44-46 ("The trusted repository can either be an expansion ROM board, not shown, that contains verified copies of the required software *or* it can be a network host 254.") (emphasis added). Apple's construction would exclude the network host embodiments.

E. "verifying the integrity"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"verifying the integrity" [Claims 1, 4]	"checking the integrity using a cryptographic hash"	"confirming the expected condition"

Rembrandt proposes that "verifying the integrity" means "checking the integrity using a cryptographic hash." Apple's construction fails to provide any clarity for this term and omits the cryptography used in the integrity verification checks taught in the '678 Patent.

The specification describes "integrity checks" that are "provided by the use of public key cryptography, a cryptographic hash function, and public key certificates." Ex. A at 4:35-45. The specification further discloses, "The present invention does this by constructing a chain of integrity checks" where the "integrity checks compare a computed cryptographic hash value with a stored digital signature associated with each component." Ex. A at 4:40-43; *see also* Ex. A at 6:14-16, 11:39-41, 20:21-25. The integrity checks also may use a cryptographic hash in the form of a modification detection code. Ex. A at 10:9-11, 20:25-27. Rembrandt derived its proposed construction directly from the specification.

In addition, the patentees distinguished two references because they failed to teach verifying the system BIOS using cryptographic hashes. Ex. A at 2:52-64. Those references provided verification through non-cryptographic techniques, such as a checksum or cyclic redundancy check (CRC). *Id.* Similar statements were made during prosecution to distinguish the claims over prior art: "the Jablon patent does not perform a verification of the system BIOS (beyond the 8-bit additive checksum which is easily forged)." Ex. B at REMBAPPL00021703.

Rembrandt proposes defining the term "verifying" as "checking," as the specification uses "check" interchangeably with "verify." *See* Ex. A at 11:26-27 ("[S]ystem integrity is preserved through the chain of *integrity checks* in the bootstrap process.") (emphasis added), 11:27-29 ("The ideal authentication chain is produced by each layer *verifying* the next . . . ."), 4:40-45 ("The present invention does this by constructing a chain of integrity checks, beginning at power-on and continuing until the final transfer of control from the bootstrap components to the operating system itself. The integrity checks compare a computed cryptographic hash value with a stored digital signature associated with each component."). Similarly, during prosecution the patentees distinguished certain prior art based on "integrity checks" required by the claims, which recite "verifying the integrity." Ex. B at REMBAPPL00021704.

Apple's proposed construction lacks support in the intrinsic record and will not aid the jury in understanding the scope of the claims. Neither "confirm" nor "confirming" appear in the specification, nor does "expected condition." Moreover, the concept of "confirming" in Apple's construction requires an expected result, whereas "verifying" does not presuppose or assume any result. Apple's proposed construction adds unnecessary uncertainty to the claims, defeating the purpose of claim construction.

F. "means for verifying the integrity of said boot components and said system BIOS"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"means for verifying the integrity of said boot components and said system BIOS" [Claim 1]	**Function**: "verifying the integrity of said boot components and said system BIOS" **Structure**: "a verification function that performs a cryptographic hash of a layer (e.g., steps 264, 272, 284, and 290 in Figs. 2b-2d) and compares the result to the value obtained from a stored signature for the layer (e.g., steps 266, 274, 286, and 292 in Figs. 2b-2d), and equivalents thereof"	**Function:** "verifying the integrity of the boot components and the system BIOS wherein integrity failures of the boot components are automatically replaced from the trusted repository" **Structure:** "Fig. 2a #254, 256; Figs. 2b-2d steps # 262, 263, 264, 266, 272, 274, 284, 286, 290, 292; Col. 10:44-46; Col. 10:47-67; Col. 11:1-8."

Both parties agree that the "means for verifying the integrity of said boot components and said system BIOS" is a means-plus-function claim element governed by 35 U.S.C. § 112 ¶ 6. Construing a means-plus-function claim element involves two steps. The first step is to define the function of the claim limitation, and the second step is to identify the corresponding structure in the specification for that function. *Golight, Inc. v. Wal-Mart Stores, Inc.*, 355 F.3d 1327, 1333-34 (Fed. Cir. 2004).

The parties agree that the function corresponding to this claim element includes "verifying the integrity of said boot components and said system BIOS" as Rembrandt proposes. Apple seeks to have the function include "wherein integrity failures are recovered through said trusted repository," which is a separate and distinct claim limitation following the means-plus-function claim element. Apple also rewrites the "wherein" clause to add "automatically" before "replacing boot components." This is legally improper. *See, e.g.*, *Golight*, 355 F.3d at 1333 ("The court must construe the function of a means-plus-function limitation to include the limitations contained in the claim language, and only those limitations." (quoting *Cardiac Pacemakers, Inc. v. St. Jude Med., Inc.*, 296 F.3d 1106, 1113 (Fed. Cir. 2002))).

The two limitations, "means for verifying the integrity of said boot components and said system BIOS" and "wherein integrity failures are recovered through said trusted repository" are separate, and they were originally separated by a comma, Ex. B at REMBAPPL00020820, which appears to have been inadvertently dropped during prosecution, Ex. B at REMBAPPL00021700. During prosecution, moreover, the patentees and patent examiner treated the "means for verifying" and "wherein" limitations separately. *See, e.g.*, Ex. B at REMBAPPL00021541, REMBAPPL00021700-05.

The "verifying" phrase is a gerund phrase defining a function, which is the traditional form of means-plus-function elements. Stephen Winslow, *Means for Improving Modern Functional Patent Claiming*, 98 GEO. L.J. 1891,1893 (2010) (noting that a means-plus-function "element will usually take the form 'means for [gerund]'or '[gerund] means,' where the gerund states the function at issue—for instance, 'means for displaying'"). The "wherein" clause is not part of any gerund phrase, and not only does it not define a function, it is an adjective clause that only makes sense as modifying the entire system. It certainly does not modify the "verifying" function because recovery is different from verification. As such, the wherein clause should not be part of the function of the means-plus-function term. *See, e.g.*, *BBA Nonwovens Simpsonville, Inc. v. Superior Nonwovens, LLC*, 303 F.3d 1332, 1343-44 (Fed. Cir. 2002) (rejecting proposed construction of the function of a means-plus-function element because it would have included a separate claim element not subject to 35 U.S.C. § 112 ¶ 6).

The specification, moreover, describes functions and structures corresponding to the verifying and recovery aspects of the invention separately, consistent with Rembrandt's proposed constructions. The specification discloses different verification and recovery "parts" of the disclosed secure-boot process. "The first is that no code is executed unless it is explicitly trusted or its integrity is verified prior to use. The second is that when an integrity failure is detected, the recovery process can recover a suitable verified replacement module." Ex. A at 6:28-33.

Because claim 1 provides no indication that the "means-plus-function" limitation requires a single means for both verifying and recovering, and the specification treats these as two separate functions implemented using different structures, there is no reason to read the "wherein" clause as further limiting the means-plus-function limitation as Apple proposes.

Regarding the corresponding structure for this element, Rembrandt proposes the structure is a "verification function that performs a cryptographic hash of a layer (e.g., steps 264, 272, 284, and 290 in Figs. 2b-2d) and compares the result to the value obtained from a stored signature for the layer (e.g., steps 266, 274, 286, and 292 in Figs. 2b-2d), and equivalents thereof," without the unnecessary inclusion of additional structures related to the separate recovery aspects of the invention. The specification also explains that the "verifying" function is performed using a cryptographic hash function and comparing a computed cryptographic hash value with an expected value. *See, e.g.*, Ex. A at 39-41. This verification function is described, for example, in Section II.A ("AEGIS BIOS Modifications") in the specification. In contrast, the recovery process is described in different sections II.B (Integrity Policy / Trusted Repository) and III (AEGIS Network Recovery Protocol) of the specification. The disclosed recovery aspects employ different functions and structures than the verification aspects, such as implementations of a trusted repository, locations of a trusted repository, how to communicate with a trusted repository, and how components are recovered from a trusted repository. Apple's proposed construction for the function of this element and its identification of the structure for this element are inconsistent with at least these aspects of the specification.

G. "wherein integrity failures are recovered through said trusted repository"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"wherein integrity failures are recovered through said trusted repository" [Claim 1]	"wherein replacement components are obtained from a trusted repository"	[part of "means for verifying" element above and construed to mean "wherein integrity failures of the boot components are automatically replaced from the trusted repository"]

Rembrandt's construction of the "wherein" clause tracks the plain meaning of the claims. This clause concerns how recovery takes place, i.e., through said trusted repository. To that end, replacement components are obtained from the trusted repository as proposed by Rembrandt.

The specification supports Rembrandt's construction. It references the recovery of integrity failures synonymously with the recovery of replacement components. The specification states that "AEGIS ROM 256 . . . contains code that allows the secure recovery of any integrity failures found during the initial bootstrap." Ex. A at 10:51-53. In the same passage, the specification explains, "If the component that fails its integrity check is a portion of BIOS 112, then [the portion of BIOS 112 that failed its integrity check] must be recovered from AEGIS ROM 256." Ex. A at 10:56-58. "A failure beyond BIOS 112 causes the system to boot into a recovery kernel" that "contacts a 'trusted' host" in order "to recover a verified copy of the failed component." Ex. A at 10:61-65. As shown above, the specification uses "recovery of any integrity failures" as shorthand for recovery of the part of BIOS or boot component(s) that failed their integrity check. Rembrandt's proposed construction reflects this synonymous usage. *See also* Ex. A at 4:48-51 ("Once an integrity failure is detected, the invention uses a secure protocol to inform a trusted repository that a failure has occurred and to obtain a valid replacement component”), 20:27-30 (same), 6:31-33 (“the recovery process can recover a suitable verified replacement module”).

Apple’s proposed construction is inconsistent with the specification. For example, Apple’s construction only encompasses recovery of boot components and does not take into account that the specification discloses recovery of both part of the BIOS and boot components. *See, e.g.*, Ex. A at 10:56-67. Apple’s proposed construction also attempts to rewrite the claim’s recitation of “recovered” to “automatic replacement,” even though the claim does not recite “automatic.” This is improper. The Federal Circuit “do[es] not read limitations from the specification into [the] claims.” *Thorner*, 669 F.3d at 1366. This is so even when “the only embodiments, or all of the embodiments, contain a particular limitation.” *Id.* Here, although the specification describes an embodiment using automated recovery, *see, e.g.*, Ex. A at 6:21-25, it also describes recovery more generally, without regard to whether it is performed automatically. *See, e.g.*, Ex. A at 10:19-25. It is incorrect to add a disclosed but unclaimed limitation of “automatic” recovery into the claims. If the patentees intended to limit the claims to “automatic” recovery, they would have included that word in the claims.

H. “a host computer”

Terms & Claims	Rembrandt’s Proposal	Apple’s Proposal
“a host computer” [Claim 3]	“a server computer attached to a network”[3]	“a separate computer system connected to the computer through a communications interface”

The claims and specification support Rembrandt’s proposed construction. Claim 3 recites “said trusted repository is a host computer communicating with said computer system through a communications interface coupled to said expansion bus.” This means the repository is a network host, which the specification describes. *See, e.g.*, Ex. A at 10:44-46 (“The trusted repository can either be an expansion ROM board, not shown, that contains verified copies of the required software or it can be network host 254”), 7:50-53, Fig. 2a. The specification also discloses that a trusted repository implemented as a network host is a server computer. *See, e.g.*, Ex. A at 17:32-34 (“Client 410 (AEGIS) and *Server 420 (Trusted Repository)* wish to communicate and establish a shared secret after authenticating the identity of each other.”) (emphasis added), Figs. 4 & 6. Thus, the intrinsic evidence shows the claimed “host computer” is a server computer attached to a network, as Rembrandt proposes.

In contrast, Apple’s proposed construction for “host computer” has no support in the specification. Apple’s construction redefines claim 3’s recitation of the “host computer communicating with said computer system” to a requirement that the host computer must be “connected to the computer” system. Nothing in the specification or prosecution history supports such a restriction. Doing so is contrary to the case law. *See, e.g.*, *Thorner*, 669 F.3d at 1365-66.

[3] In the joint claim construction statement, Rembrandt proposed that “host computer” means “a computer, attached to a network, providing primary services such as computation, data base access or special programs or programming languages.” Because the ’678 Patent refers to a computer that provides primary services to other computers (clients) as a “server” computer, *see, e.g.*, Ex. A at Figs. 4 and 6, Rembrandt has shortened its proposed construction to clarify that the claimed “host computer” is a server computer attached to a network.

I. “Power on Self Test (POST)”

Terms & Claims	Rembrandt’s Proposal	Apple’s Proposal
“Power On Self Test (POST)” [Claim 4]	“one or more diagnostic tests performed by the computer system during its startup”	“one or more diagnostic tests performed by the system BIOS upon system startup”

The parties agree that the claimed “Power On Self Test (POST)” refers to “one or more diagnostic tests” performed during startup of the computer system. The parties dispute, however, whether these diagnostic tests are limited to tests “performed by the system BIOS,” or are more generally “performed by the computer system.”

The claims and specification do not limit POST to diagnostic tests “performed by the system BIOS” as proposed by Apple. Claim 4 simply requires “invoking a Power on Self Test (POST),” and contains no limitation regarding the component that invokes or performs the POST. The specification lists four ways to invoke the POST. Ex. A at 7:62-8:37. It also explains that the initial processor self-test is not under the control of system BIOS. Ex. A at 8:10-11. Moreover, the preferred embodiment initiates POST before executing the first layer of system BIOS software. For example, Figures 2b-2d each have a block labeled “Initiate POST” that is performed before the first block indicating action by the system BIOS.

Rembrandt’s proposed construction is in accord with the claims and specification because it does not have the limitation Apple seeks to add. Apple’s construction excludes portions of the preferred embodiment, and claim interpretations that exclude preferred embodiments are seldom correct. *See, e.g., On-Line Techs. v. Bodenseewerk Perkin-Elmer Corp.*, 386 F.3d 1133, 1138 (Fed. Cir. 2004).

J. “when said boot component fails, recovering said failed boot component” and “to replace said boot components”

Terms & Claims	Rembrandt’s Proposal	Apple’s Proposal
“when said boot component fails its integrity verification, recovering said failed boot component . . . to replace said failed boot component.” [Claims 4, 7]	“when said boot component fails its integrity verification, recovering said boot component . . . to replace said boot component that failed its integrity verification”	“at the time boot component fails its integrity verification, automatically recovering said failed boot component . . . automatically replacing said failed boot component.”

1. “when said boot component fails, recovering said failed boot component”

The parties agree that this term applies to a “boot component fail[ing] its integrity verification.” The parties dispute (i) whether “when” means “at the time”; (ii) whether this claim term requires “automatically recovering”; and (iii) whether a “boot component” or a “failed boot component” is recovered.

Regarding issue (i), the specification discloses a two-step recovery process. First, an integrity failure of a boot component must occur, then recovery of the boot component occurs later. The Federal Circuit has acknowledged that “when” has several meanings, depending on the context. *See, e.g., Renishaw PLC v. Marposs Societa Per Azioni*, 158 F.3d 1243, 1251 (Fed. Cir. 1998) (observing that “when” can mean “at or during the time that; just at the moment that; at any or every time that; [or] at, during, or after the time that”). Here, the specification does not support defining “when” as “at the time,” as Apple proposes. Instead, the specification describes a sequence of events in which recovery occurs after detecting an integrity failure. For example, the specification states, “Once an integrity failure is detected, the invention uses a secure protocol to inform a trusted repository that a failure has occurred and to obtain a valid replacement component." Ex. A at 20:27-30; *see also* Ex. A at 4:48-51 (same).

Figure 2b, excerpted and annotated below, shows performing a recovery process at step 298, which is after a boot component has failed its integrity check at step 286:

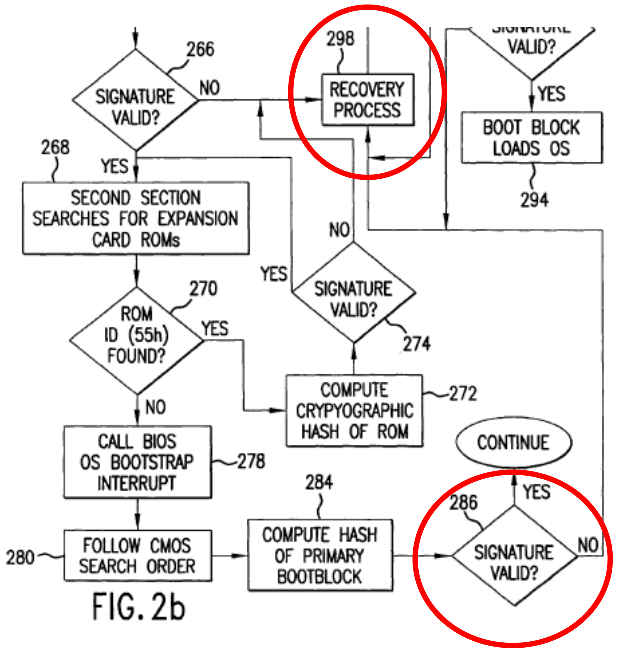

FIG. 2b

The specification does not support construing "when" as "at the time."

Regarding issue (ii), Apple again seeks to insert "automatically" into the claims. As explained above, *supra* Part IV.G, the claim does not recite "automatically recovering," and there is no reason to read "automatically" into this claim term. The patentees did not define recovery as being limited to "automatic" recovery, nor is there any clear and unmistakable disclaimer in the intrinsic evidence that would disclaim recovery processes that are not automatic. *See, e.g.*, Ex. A at 10:19-25.

With respect to issue (iii), the parties dispute whether the "boot component" or the "failed boot component" is recovered. Setting aside the issue of whether "when" means "at the time," the parties agree that "when said boot component fails" means "when said boot component fails its integrity verification." This construction makes clear that the antecedent basis for the "boot component" is the "boot component that failed its integrity check," as Rembrandt's construction provides.

The specification also describes obtaining a replacement of the boot component that fails its integrity verification. *See*, *e.g.*, Ex. A at 4:48-51, 20:27-30. Replacing a failed boot component with another failed boot component, as Apple's proposed construction requires, does not result in recovery. As a result, the Court should adopt Rembrandt's proposed construction.

2. "to replace said failed boot component"

Rembrandt proposes construing "to replace said failed boot component" as "to replace said boot component that failed its integrity verification." The parties dispute whether (i) the term should require "automatic replacement"; and (ii) whether "the boot component that failed its integrity verification" or the "failed boot component" is replaced. As explained above, it is improper to read "automatically" into the claim, *see supra* Part IV.J.1 and Part IV.G, and the "failed boot component" is "the boot component that failed its integrity verification" and not another failed boot component, *see supra* Part IV.J.1. This claim term appears in claim 7, which depends from independent claim 4, and modifies claim 4's recitation of "recovering said failed boot component." Ex. A at 22:25-26, 22:37-40.

K. "secure protocol"

Terms & Claims	Rembrandt's Proposal	Apple's Proposal
"secure protocol" [Claim 7]	"cryptography combined with a protocol to add security to the recovery process"	"a standard enabling communication between the computer system and the trusted repository secured through a cryptographic algorithm"

The parties agree that the claimed "secure protocol" uses cryptography to provide security. The parties disagree, however, whether the secure "protocol" is limited to "a standard enabling communication between the computer system and the trusted repository" as Apple proposes.

Rembrandt's proposed construction for "secure protocol" comes directly from the specification, which states, "The secure protocol of the present invention can be based on well known networking protocols . . . or on a custom protocol," Ex. A at 4:51-55, and "Cryptographic algorithms are combined with the chosen protocols to add security to the recovery process," Ex. A at 4:56-57. *See also* Ex. A at 20:31-35. The specification requires the claimed "secure protocol" be "cryptography combined with a protocol to add security to the recovery process," as Rembrandt proposes.

Rembrandt's construction also aligns with the exemplary "secure protocol" used in the AEGIS embodiment. The specification discloses basing the secure protocol on the Dynamic Host Configuration Protocol (DHCP) and Trivial File Transfer Protocol (TFTP). Ex A at 4:53-54. "[A]uthentication is added to DHCP and TFTP [so] AEGIS can use them without further modifications as its recovery protocol." Ex. A at 19:36-38. Thus, the "protocol" in the AEGIS embodiment is based on DHCP and TFTP. *See, e.g.,* Ex. A at 4:41-56, 18:31-19:45. The "cryptography" added to the protocol in the AEGIS embodiment is the authentication added to DHCP and TFTP, for example, by adding SHA1 message digests (cryptographic hash values) to DHCP and TFTP messages. *See* Ex. A at 15:50-67, 18:24-27, 19:38-45.

Apple's proposed construction finds no support in the specification. Apple equates the claimed "protocol" with "a standard enabling communication between the computer system and the trusted repository," but the specification teaches "the secure protocol of the present invention can be based . . . on a custom protocol." Ex. A at 4:51-55. Something that is "custom" is, by definition, not standard. Apple’s proposed construction based on a “standard” protocol excludes use of a custom protocol as taught in the specification. It cannot be proper.

V. CONCLUSION

Rembrandt requests that the Court adopt its proposed constructions, which are consistent with the claim language, specification, prosecution history, and the law of claim construction.

Dated: September 24, 2014

Respectfully submitted,

/s/ Trey Yarbrough
Trey Yarbrough
Bar No. 22133500
YARBROUGH WILCOX, PLLC
100 E. Ferguson St., Ste. 1015
Tyler, TX 75702
(903) 595-3111
Fax: (903) 595-0191
trey@yw-lawfirm.com

Gerald F. Ivey (*pro hac vice*)
E. Robert Yoches (*pro hac vice*)
Richard B. Racine (*pro hac vice*)
Christopher T. Blackford (*pro hac vice*)
FINNEGAN, HENDERSON, FARABOW
GARRETT & DUNNER, LLP
901 New York Avenue, N.W.
Washington, D.C. 20001
(202) 408-4000

Stephen E. Kabakoff (*pro hac vice*)
Anita Bhushan (*pro hac vice)*
Benjamin Schlesinger (*pro hac vice*)
FINNEGAN, HENDERSON, FARABOW
GARRETT & DUNNER, LLP
3500 SunTrust Plaza
303 Peachtree Street, N.E.
Atlanta, GA 30308-3263
(404) 653-6400

Jacob Schroeder (*pro hac vice*)
FINNEGAN, HENDERSON, FARABOW
GARRETT & DUNNER, LLP
3300 Hillview Avenue
Palo Alto, CA 94304
(650) 849-6600

ATTORNEYS FOR PLAINTIFFS
REMBRANDT PATENT INNOVATIONS, LLC and REMBRANDT SECURE COMPUTING, LP

CERTIFICATE OF SERVICE

The undersigned hereby certifies that all counsel of record who are deemed to have consented to electronic service are being served with a copy of this document via the Court's CM/ECF system per Local Rule CV-5(a)(3) on this 24th day of September, 2014. All other counsel not deemed to have consented to service in such manner will be served via facsimile transmission and/or first class mail.

/s/ Trey Yarbrough
Trey Yarbrough